AD-A037 894 RAND CORP SANTA MONICA CALIF F/G 9/2

LIMDEP - A REGRESSION PROGRAM FOR LIMITED DEPENDENT VARIABLES,(U)

JUL 76 C E PHELPS

UNCLASSIFIED P-5696 NL

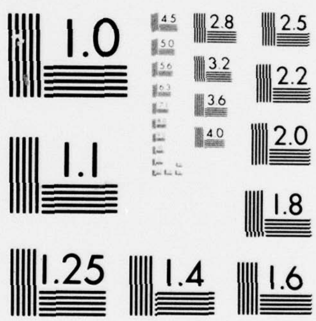
1 OF 1

AD A037894

END

DATE FILMED

4-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

2 B.S.

6

LIMDEP - A REGRESSION PROGRAM FOR LIMITED DEPENDENT VARIABLES

10

Charles E. Phelps

11

Jul[redacted]76

12

23 p.

14

P-5696

296 600

mts

blank
ii

The Rand Paper Series

TABLE OF CONTENTS

LIMITED DEPENDENT VARIABLE REGRESSION PROGRAM--LIMDEP

Charles E. Phelps

July 1976

INTRODUCTION

LIMDEP (Version Phelps) is a Fortran IV (System 370) program for performing regressions using the limited dependent variable technique. The program is wholly double precision, although each observation is read in as a single precision vector, thus permitting the use of F and E formats. Provision is made in the program for performing regressions with the observations weighted. Weights are automatically normalized. Data may be supplied either on cards or on tape or disk input. Each observation consists of N words, where: words 1 through N-3 are the independent variables, word N-2 is the dependent variable, word N-1 is the limit value, and word N is the weight for the observation. If the user wishes to perform regressions with unweighted observations, the Nth word may be left blank, and the weight will automatically be set to 1.

The limited dependent variable technique may be appropriate when the dependent variable for the t-th observation (t = 1, 2, ..., T) cannot fall below (or above) some finite value L. Thus, the range of the dependent variable for the t-th observation is:

$$L \leq y_t < \infty \text{ (or } -\infty \; y_t \leq L \text{ if the limit is an upper limit).}$$

The basic model assumes that an index I is given by

$$I = X_1B_1 + X_2B_2 \ldots + X_kB_k.$$

Behavior is not fully explained by this "model," the differences in behavior being attributed to a random variable ε, distributed $N(0,\sigma)$. Let y be the observed dependent variable. Then behavior is assumed to be as follows:

$$y = L \qquad I - \varepsilon \leq L$$
$$y = I-\varepsilon \qquad I - \varepsilon \geq L$$

The program (LIMDEP) selects the set of coefficients $\hat{B}_i$ which maximize the likelihood of the data, given the underlying model. The expected value of y (E(y)) can be computed directly from the coefficient vector for any vector X_t. A graphical representation is shown in Fig. 1. A detailed exposition of the limited dependent variables model may be found in "Estimation of Relationships for Limited Dependent Variables," by James Tobin, in Econometrica, Vol. 26, No. 1, 1958, pp. 26-36.

The program will accommodate one dependent variable and up to 60 independent variables. [Note that the data source may contain more than one dependent variable and 60 independent variables, but actual reading of the data (via FORMAT statement) must limit the number of variables transferred to the program to the above stated limits.]

OUTPUT OF THE PROGRAM

The output of the program includes eight sections, each described here:

(1) Second moments of all the variables around the origin for the limit observations, and for the non-limit observations, as well as means, standard deviations, and maximum and minimum values of all variables. These data summaries are given separately for limit, non-limit, and all observations. The weighted and non-weighted number of observations is also given for each of those three categories. In addition, the first 10 included observations are also listed to allow the programmer to further check his input data. [NOTE: The program presently deletes all observations in a weighted regression which are assigned a weight of zero. These observations are treated as never having existed on the original data source -- they do not enter the moment matrix or the observation "count." The number of observations so deleted may be computed by comparing the unweighted total number observations to the original number (NOB) specified on the control card (card 1 of the data section).]

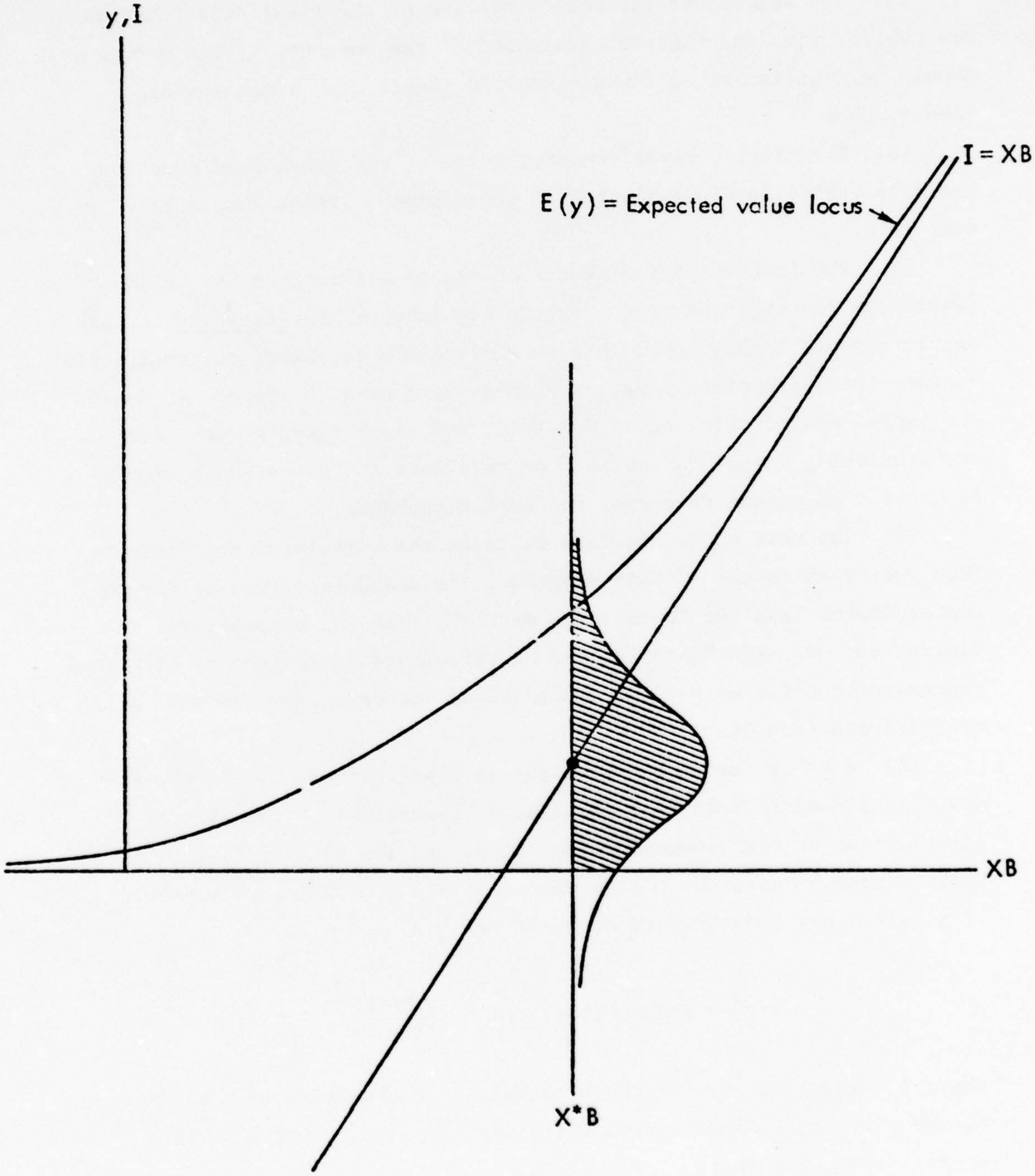


Fig. 1 —The total expected value locus (shaded area shows $P(y>0 \mid X^*, B)$

(2) The second derivatives of the log of the likelihood function are tabled, with the function evaluated at the maximum. (This matrix should be negative along the diagonal to insure that a maximum has been reached.)

(3) The first derivatives of the log of the likelihood function are also tabled, again evaluated at the maximum. These should be very near zero.

(4) The inverse of the matrix of second derivatives of the log-likelihood function are given next in the output. The negative of this matrix asymptotically approaches the covariance matrix of the coefficient vector--the appropriate diagonal elements are used to compute "t-ratios" for regression coefficients. THESE ARE NOT exact t-tests--they are asymptotically normal variables; the reference to "t-tests" is simply to provide an analog to normal regression packages.

(5) The next output section contains the normalized coefficients (the vector AM in the FØRTRAN program), the standard errors of the AM vector (taken from the "covariance matrix"), and the unnormalized coefficient vector. The "unnormalized" coefficients correspond to the usual regression coefficients--they should be of approximately the same scale as coefficients from OLS.

(6) A table for each regression is also printed, giving the non-normalized coefficients, the associated "t-statistic," the partial elasticities of the expected value locus, and the elasticities of the tobit "index," computed at mean values of all X_i. The expected value of y, given any X-matrix, is computed as:

$$E(y) = P \cdot X\beta + (1-P) \cdot (L) + \sigma f\left(\frac{X\beta - L}{\sigma}\right)$$

where L = Limit value, P = the probability of observing (y > L) given X, and σ is the estimated standard error around the "index." It is easily shown that where

$$\beta_i = \frac{\partial I}{\partial X_i}$$

then

$$\frac{\partial E(y)}{\partial X_i} = P \cdot \beta_i$$

so that

$$\eta_{E(y),x_i} = P \cdot \beta_i \frac{X_i}{E(y|X)}$$

and

$$\eta_{I,X_i} = \beta_i \frac{X_i}{y}$$

The table gives these elasticities at mean values of all X_i. Also given in this table is the mean squared error of the predictions (using each observation's expected value), the mean error, and the predicted and observed frequencies of non-limit cases. The maximum likelihood method does not guarantee unbiased predictions on y (the dependent variable), unless the error term in the data is exactly normal. Most users of the program find that their predictions have non-zero mean error. Because of this bias in the expected value of y, the mean-squared error may in fact be larger for tobit regressions than for OLS regressions.

(7) (This section may be deleted at the user's option, by setting ISUP on the control card appropriately. The reasons will become apparent shortly.) This section contains, for each observation, a list of:

(a) z_t = the distance in "σ" units that the observation lies from the limit value.

(b) The probability that the observation exceeds the limit value. When the value in (a) is equal to zero, the value of (b) is 1/2. When z_t is negative, (b) is less than .5, and when z_t is positive, (b) is greater than .5. In other words, z_t is a "normal distribution" z-score, and the value of (b) is the cumulative probability of observing that z.

(c) The normal density as z_t. (This is necessary to compute the expected value of the dependent variable.)

(d) The expected value of the dependent variable. See above for the appropriate formula. Note that the expected value is always positive.

(e) The observed value of the dependent variable.

(f) The limit value for the dependent variable. Note that there may be a different limit for each observation.

Since this section contains an output list as long as the number of observations, the programmer may wish to delete this section. (The comparable output would be a plot of residuals in a normal OLS run.) For large NOBS, this section may be tedious and of little use. Hence, the user has been given the opportunity to suppress this section of the output through ISUP.

(8) The log of the maximum value of the likelihood function is given for each regression. In computing the likelihood function, if z_t is greater than 12, it is truncated to equal 12.0. This value may be used to compute a chi-square statistic testing the proposition that $b_1 = b_2 = \ldots = b_k = 0$ for this regression, if the appropriate steps are taken. For each dependent variable, the user should run a separate regression with only the constant term included on the explanatory variable list. This is the "constrained" likelihood function. Then the difference between the log-likelihood of the constrained function and the log-likelihood of the unconstrained function (i.e., your regression) is λ, the log of the likelihood ratio. For large n, -2λ is distributed chi-square with k degrees of freedom, where k is the number of explanatory variables in your regression (not including the constant term). You cannot compute this chi-square statistic unless you make the provision of running the separate regression for each dependent variable with only the constant term. (Similarly, the chi-square test may be used to test the significance of any variable, or any subset of variables, by running a "partly constrained" regression, and following the same procedure. This is analogous to an F-test on a vector of coefficients in standard OLS regressions.)

TRANSFORMATION OF DATA

The program is written so that a subroutine called TRANS is called immediately after each observation is read. A dummy TRANS routine is supplied, with the necessary DIMENSIØN, DØUBLE PRECISIØN, and CØMMØN statements. The user may transform his data in any desired fashion, writing his own FØRTRAN IV version of TRANS.

Note that variables may be rearranged, as well as transformed, within TRANS. This feature may allow you to use data sources which have not explicitly been designed for LIMDEP. All that is required of the data is that AFTER you have returned from TRANS, the data are arranged as suggested on the first page of this write-up: independent variables, dependent variable, limit variable, and weight. For example, if your data do not have either a limit value, a weight, or a constant term, and your dependent variable is not located as the last variable, you can rearrange and create variables appropriately in TRANS. The variables must be in the correct order upon their return from TRANS. Example: Suppose you have a tape with one dependent variable, then 10 independent variables (but no constant term, limit, or weight). Set NVR = 9 on your main problem card (the program will then read your 11 variables), and set NVS = 3. Then, you rearrange variables in the following manner:

```
X(14) = 1.0D0          This weights all obs. = 1.0
X(13) = 0.0D0          The limit = 0
X(12) = X(1)           X(12) is now the dependent variable
X(1)  = 1.0D0          X(1) is now your constant term
```

The remaining 10 variables (X(2) → X(11)) are your other explanatory variables. You would then set up your title cards so that the first "title" was "CØNST," the second was for your "first" explanatory variable, etc. The twelfth "title" would be your name for the dependent variable. You would code NK = 1 in columns 14-15 of the main problem card.

It is strongly suggested that all operations performed in TRANS be carried on in DØUBLE PRECISIØN (e.g., use DSQRT rather than SQRT, X(3) = 1.5D0, rather than X(3) = 1.5, etc.). The vector X is double

precision, and any single precision operations may carry extra bits of information that are not desired by the user. It is particularly important that any computations involving the weight term (X(NVU)) be done in double precision. Available through CØMMØN are the following variables (among others):

NVR — The number of variables read in via card, tape, or disk, not including the limit or weight.

NVS — The total number of variables after transformation (the data card originally read NVR as number of variables read in, and NVS as the number of variables added). However, the program immediately sets NVS = NVR + NVS, so that NVS is the set number of variables to be used by LIMDEP. Note that (on the original control card), NVS may be negative, so that a data tape containing more than the desired number of variables may be read, and some may be deleted through the TRANS routine.

NVT — The index of the LIMIT value (i.e., X(NVT) is the limit value).

NVU — The index of the WEIGHT value (i.e., X(NVU) is the weight used in weighted regressions). Note well: if X(NVU) is set (or read in) as 0.0 (double precision), and if you specify on the main control card that you are not performing weighted regressions, then the weight will automatically be set to 1.0 (double precision). If you specify that the weight term is equal to 0.0 (double precision) and that you are performing weighted regressions, then that observation will be deleted upon returning to MAIN from TRANS. That observation will enter neither the moment matrix nor the count of weighted or unweighted observations.

You should also be aware that, in weighted regressions, the weights are normalized, so that the sum of the weights (after normalization) is equal to the unweighted number of observations.

NK — The index of the variable which is the constant term in your data array.

NOB — The number of observations to be read from your data source.

Also available are double precision constants for π and $1/\sqrt{2\pi}$ (PI and RTP, respectively).

Note well: If a constant term is desired in your regressions, you must explicitly include one in your data array. If your data source does not include a vector of constants, it is suggested that you add such a variable through the TRANS routine, simply by specifying that X(NK) = 1.0D0 for all observations.

DATA SECTION OF THE PROGRAM

There may be as many data packs in each program run as the user desires. A separate data pack must be included for each dependent variable upon which regressions are being computed. (The reason for this is that the program sorts the data onto two scratch disks, depending on whether the dependent variable is at the limit or exceeds the limit. Obviously, as you change dependent variables, you of necessity change the arrangement of some of the observations. Therefore, it is necessary to re-read the data and sort according to the new dependent variables' value.) You may run as many regressions as you desire on each dependent variable. That is, there may be NJB regressions in each data pack. (Recall from the discussion of the chi-square test that it may be valuable to include one regression with just the constant term in each data pack.) Equations for different dependent variables may be estimated in the same computer run by re-reading your data tape several times (a different data pack for each pass through the tape), using different FØRMAT statements in each data pack. That is, your original data source may indeed contain more than one dependent variable, but the LIMDEP program can only process them one at a time. By using the feature by which tapes can be read with or without rewinding, you can get to the appropriate location on your data tape to reprocess the observations anew, with new dependent variables read for each data pack. Similarly, data read in from cards may be processed sequentially, so that multiple dependent variables can be studied in the same computer run. There are frequently significant economies to running long jobs on the 370 (for example, at Rand there is a block time rate for overnight jobs which has a decreasing marginal cost for longer computer runs); the user is urged to investigate these possibilities fully--LIMDEP is an

expensive mode of analysis otherwise, and cost-saving methods are possibly quite effective.

Each data pack contains the following four sections:

A. Card 1:

Cols. 1-3 (NJB) The total number of regressions to be performed in this data pack (i.e., on this dependent variable).

Cols. 4-6 (NVR) The number of variables read in from the data source for each observation, including the dependent variable (but not including the limit or weight forms).

Cols. 7-9 (NVS) Net increase in the number of variables after transformation (using the TRANS routine). This variable may be negative; it will equal zero if the TRANS routine is not used.

Cols. 10-13 (NOB) The number of observations to be read from the data source. The limit is 9999. (To increase the limit further, one must simply alter the FØRMAT card in MAIN, and make commensurate adjustments in the location of all variables which follow NOB on this card.)

Cols. 14-15 (NK) Index of the variable which is the constant term. This variable must be supplied if summary statistics are desired. (i.e., if MSUP [col. 22 of this card] is left at zero [the default value], then a value for NK must be supplied.)

Col. 16 (NA)
Set = 0 if no additional data packs follow this one.
Set = 1 if there are additional data packs.

Col. 17 (NB)
Set = 0 if the limit value is a lower limit.
Set = 1 if the limit value is an upper limit.

Col. 18 (ISUP)
Set = 1 if you wish to suppress the listing of the six values for each observation discussed as Section 7 of the output of LIMDEP.
Set = 0 if you desire that table. (It is recommended that ISUP = 1 be used for very large numbers of observations.)

Col. 19 (ITAPE)
Set = 0 if your data pack contains your observations on cards. If ITAPE = 0, your data pack must contain NOB data cards. (See discussion below.)

Set = 1 if your data input is via TAPE or DISK, and you wish that device to be rewound before it is read.

Set = 2 if your input is via TAPE or DISK and you DO NOT wish that device to be rewound before using.

Cols. 20-21	(NIT) The maximum number of iterations to be permitted. Default value (if you leave these columns blank) is 25 iterations. Typically, LIMDEP will finish a regression by using 4-10 iterations.
Col. 22	(MSUP) Set = 1 if the data summary section is not desired (if MSUP is left blank or explicitly coded to zero, then you must supply a value for NK, and you must have a constant term (= 1 for all observations) in your data set.
Col. 23-24	(NWT) Set = 01 if the regressions in this data pack are to be weighted regressions. Default is zero (unweighted, in which case the weight will automatically be Set = 1 for each observation if you have not given the weight an explicit value.
Cols. 25-26	(NFMT) The number of FØRMAT cards which follow.
Cards 2-***	Immediately following are NFMT cards, containing the format by which the program will read your data for this data pack. If you have NFMT = 2, then these cards will be Cards 2-3. If NFMT = 1, then this will be Card 2 only (etc.). The FØRMAT cards must contain a format statement which will account for (NVR + 2) variables, where NVR is the number of independent and dependent variables to be read in, and two more are added for the limit and the weight. Note that the limit and the weight may be "dummy" zero values which may then be set appropriately in TRANS. If you are using tape or disk input, you must read some values in for the limit and weight--it may be necessary to create a special tape for LIMDEP runs, if your primary data tape does not have these extra two variables on it. The format should begin with a "(" --a left parenthesis--in Column 1 of the first card, and end with a ")" just after the last expression in the format statement on the last format card. You may use all 80 columns of each card. All standard FØRTRAN FØRMAT expressions are allowed.

B. **Title Cards--Use as many title cards as are necessary to write out titles for each variable you use. Up to six characters may be used for each title, with twelve titles per card. (For example, if you are reading in 25 variables, you will need three cards.) The format of each title card is as follows:**

Cols. 1-6	The name of the first variable
Cols. 7-12	The name of the second variable, etc., up to and including Column 72.

C. **The observations (if on cards), one observation at a time, which will be read in the user specified format. Obviously, if the data input is via tape or disk, then this section of this data pack will be empty.**

D. Specification of the regressions. There must be NJOB sets of these cards in each data pack (see Cols. 1-3 of the main control card), containing the following information: whether or not you are reading in an initial coefficient vector guess; whether you have any quadratic linear combinations you wish to test; how many explanatory variables there are; and the numbers of the variables included in the regression. The regression control card is of the following form:

Col. 1 — (IVCTR) Set = 1 if you are reading in an initial coefficient vector guess. If you leave this column blank (the normal procedure), then LIMDEP computes an initial coefficient vector. If you set IVCTR=1, you <u>must</u> include the appropriate number of cards for the initial coefficient vector.

Col. 2 — (IQUAD) If you have any quadratic terms in your regression, and you would like to test the appropriate linear combination of variables to find the underlying effect of the "true" variable, set IQUAD equal to the number of such quadratic tests you would like performed. For example, if you have income, income-squared, price, price-squared, and education and education-squared as explanatory variables (as well as others in linear form), then you would set IQUAD equal to 3. LIMDEP will then compute the appropriate slopes and elasticity of E(y) with respect to income, price, and education.

Since $I = b_o + b_1 \text{(Income)} + b_2(\text{Income}^2) + \ldots + b_k x_k$

Then $\partial I/\partial(\text{Income}) = b_1 + 2 \cdot (b_2) \cdot (\text{Income})$ and similarly for other variables entered in quadratic form. LIMDEP evaluates this expression at the average value of income (and other variables) to compute the slope, and then compute the elasticity of E(y) from this slope, again evaluating at average values. If you set IQUAD equal to some positive number, you <u>must</u> include a card which gives the locations of the appropriate variables. You do <u>not</u> put the <u>variable numbers</u> on that card -- you put the locations on the regression control card. An example will follow below.

Col. 3-4 (NX) The number of explanatory variables in this regression, including any constant term you wish to include, (but not including the dependent variable).

Col. 5-7 The number of the first variable in the regression, right justified.

Col. 8-10 The number of the second variable in the regression, right justified.

Cols. 11-13, etc. up to and including column 73. If you include more than 23 variables in your regression, then continue with variables 24-40 on the next card, in a similar fashion. That is, on card 2 of the regression package (optional), begin in column 1-3 with the number of the 24th variable, continuing up to NX variables.

Section 2 of the regression control (OPTIONAL). If you set IVCTR=1, then you must include sufficient number of cards, immediately following the regression control card(s) to read in NX initial coefficient guesses, in a format of (5F16.0). That is, put five of the coefficients per card, beginning in columns 1, 17, 33, 49, and 65. These must be right justified unless you punch in an explicit decimal point.

Section 3 of the regression control (OPTIONAL). If you set IQUAD to a positive number, then there must be IQUAD pairs of numbers, giving the variables on which you want the linear combinations computed for quadraticly entered variables (see discussion above relating to IQUAD). You put on this card the positions of the variables on the control card. The format used is (20I4)--use four columns per variable. Up to 10 pairs may be tested for each regression. A sample regression control packet follows:

```
0312  06 07 10 13 12 14 01 15 02 03 04 05
  01  03  06  12  10  11
```

The first card specifies that IVCTR=0 (no initial vector guess), that there are 3 quadratic tests to be performed, and that there are 12 variables total in the regression (variables 6,7,10,13,12,14,1,15,2,3,4, and 5).

The second card specifies the pairs of variables to be tested in quadratic combinations. The three combinations to be tested are the first and third variables in the regression, (variables 06 and 10), the sixth and twelfth variables in the regression (variables 14 and 05), and the ten and eleventh variables in the regression (variables 03 and 04). Note carefully that while the regression control card uses the variable numbers, the quadratic tests indicates the positions of the variables on the regression control card itself. Please be careful when using this option.

Visually, your data pack will appear as follows in Fig. 2. As many data packs as desired may be included in each job. (See Column 16 of main control card.) Your entire deck will look like Fig. 3.

RUNNING THE PROGRAM ON SYSTEM 370

Two scratch devices are necessary to run LIMDEP, which should be specified on JCL cards as LOGICAL 09 and LOGICAL 10. Be sure to allow plenty of space on the scratch devices to accommodate the entire

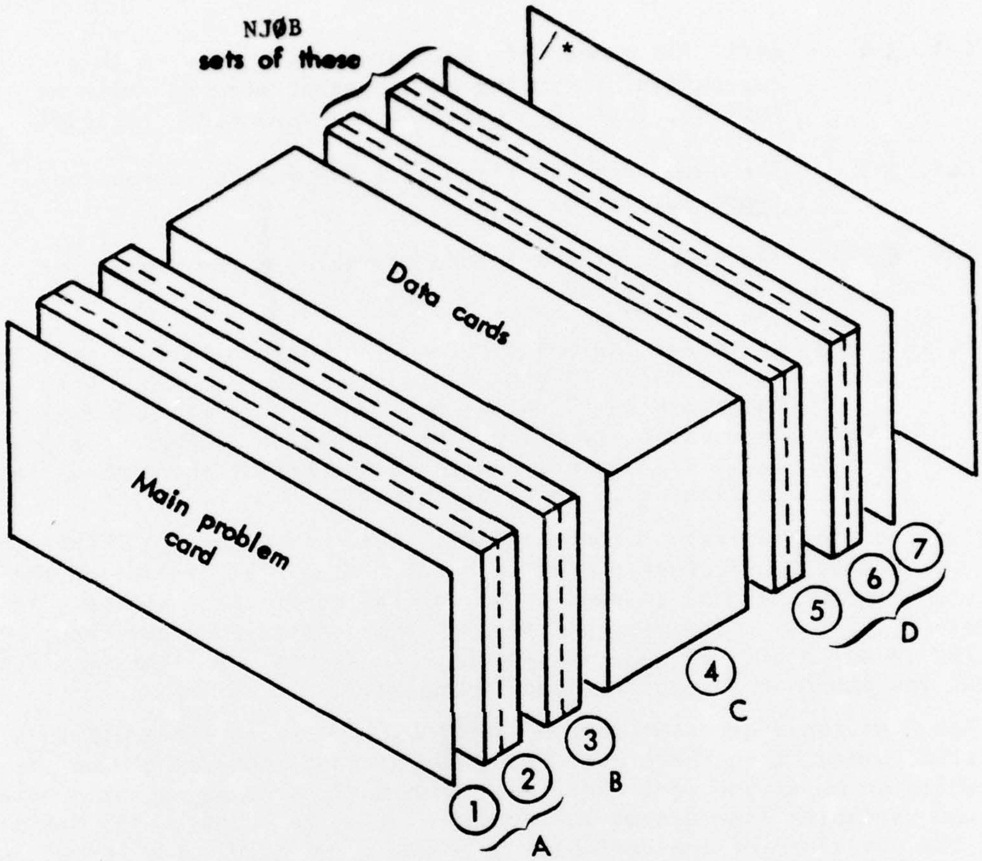


Fig. 2 —A "Data Pack"

1. Main Control Card
2. Format Card(s)--there must be NFMT of these.
3. Title Cards--use as many as necessary to give titles to all (NVR + NVS) variables (12 per card in cols. 1-72).
4. Data input (OPTIONAL)--If data input is via tape or disk, this section is empty.
5. Regression control card(s). If NX > 23, there are two of these; otherwise, this section has only 1 card.
6. Initial coefficient vector guess (OPTIONAL). Five values per card, giving NX coefficients. Use only if IVCTR = 1 on regression control card.
7. The pairs of quadratic terms being tested. Use only if IQUAD is positive. If used, include IQUAD pairs of variable locations from the regression control card. See text.

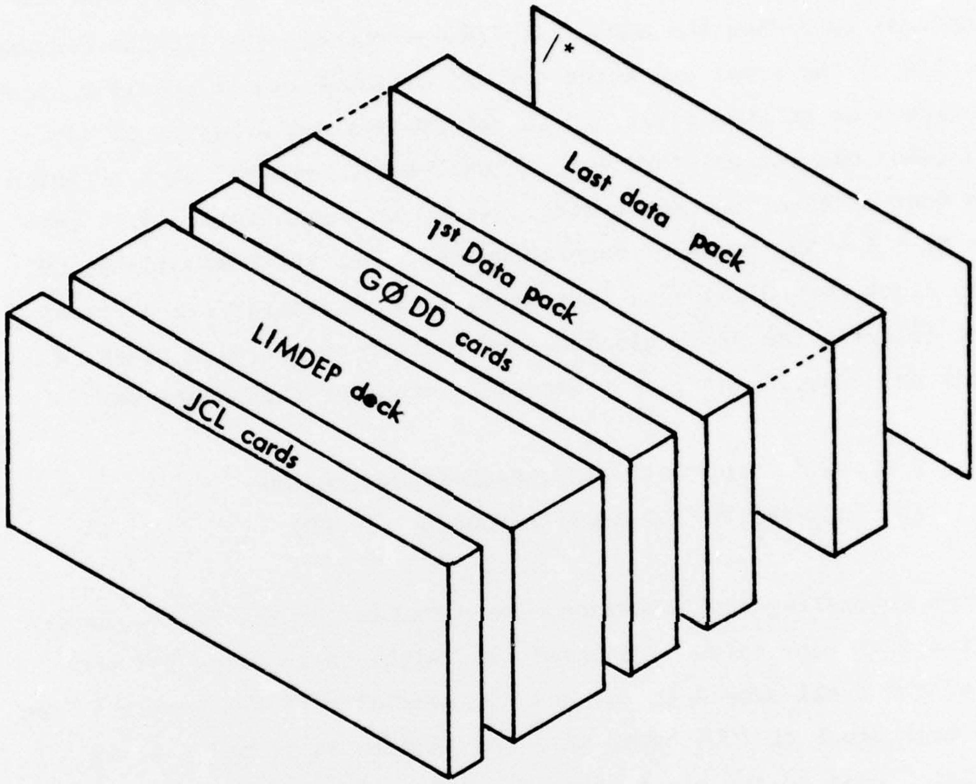


Fig. 3 —Typical LIMDEP JØB setup

1. Beginning JCL cards (job card EXEC FØRTCLG, //FØRT.SYSIN DD*, etc.
2. The LIMDEP package, including any revisions in TRANS you have made, ending with a /* card.
3. GØ step DD cards--see section entitled "Running the Program on 370."
4. FIRST DATA PACK.

5...(Additional data packs.) End with a /* card.

dataset. It is suggested that you block these devices (USING the DCB parameters) to reduce the number of I/Ø operations--the I/Ø can consume about 3/4 of the total operating cost of a LIMDEP run if poorly blocked. The program as written fills out an entire data set array of 62 variables (plus one control word for the unformatted write), each of which takes four bytes of storage space. Hence, each observation will take (62 x 8) + 4 = 500 bytes of scratch space. Your LRECL should be 500. If you block the output onto disk as 14 logical records per physical record (block), the whole block can be written on one disk track, an optimal procedure. The appropriate JCL setup for this would be:

```
//GØ.FT09001 DD UNIT=TEMP,SPACE=(7004,(nn,mm)
//    DCB=(RECFM=VBS,BLKSIZE=7004,LRECL=500)
```

When requesting scratch space, the variables nn and mm should be selected with some care. A general rule might be to allow for one-half of the total sample in nn, and the remainder of the sample in mm. Since each block of 7004 bytes will accommodate 14 records, it is relatively easy to establish requisite values for nn and mm. For example, if there are 1000 observations divided arbitrarily between limit and non-limit observations, we want to allow for 500 observations in primary allocation, so that nn = 500/14 = 36, and secondary allocation (in units of 15 blocks) of perhaps 3. Total allocation would then be 36 + 15(3) = 81 which will accommodate 1134 observations. The job will usually run once primary allocation has been obtained. For complete "protection," request all necessary disk space as "primary." This space allocation must be made on both units 09 and 10, unless you have prior knowledge about distribution of the observations between limit and non-limit allocation. It is easy to play this on the safe side--you are not charged for disk allocation requests--the only cost involved in being careful is that your job may take a bit longer to move through the job queue if you have larger disk allocation requests.

A similar card should be prepared for LØGICAL 10. If you want to save a bit on your core requirements, use BUFNØ = 1 as an additional DCB

subparameter. Otherwise, you will have to have 2 x 7004 core for each scratch device, or about 28K. The program in addition (using FØRTRAN G) will consume about 112K. Be sure to add additional core space for any input tapes or disk (allowing for double buffering, if necessary), and for your I/Ø devices. Normally, output device 6 is triple blocked, so that (a common arrangement) if 10 lines of output are outputted together (BLKSIZE = 1330), roughly 4K must be added for the unit 6.

Hence, your core requirements might be as follows:

LIMDEP	112K
FT09 (double buffered)	14K
FT10 (double buffered)	14K
FT11 (double buffered)	20K
assuming your input tape is blocked to 10,000 bytes/record	
FT06 (triple buffered--15 lines/ record)	6K
TOTAL	166K

If you single buffer these devices (i.e., use BUFNØ = 1 for FT09, FT10, and FT11) you would require 112K + 7K + 7K + 10K + 6K = 142K, which will reduce your core requirements, but increase your run-time somewhat. When running jobs on block time, always double-buffer the program, since you are not charged for the additional core space.

Using the Link-Edit step effectively once the LIMDEP program has been previously stored on disk:

The entire compiled LIMDEP program can be put on disk, thus saving significant compilation time and card read-in. It can be best stored in your personal library, under some such name as LIMDEP.

A good routine, which will conserve personal library space significantly, would be to use the following JCL:

```
// EXEC FØRTCLG,REGION.GO=nnnK
//FØRT.SYSIN DD *
```

(Include your new TRANS subroutine here. Be sure to have the appropriate CØMMØN and DIMENSIØN and EQUIVALENCE statements from the original TRANS.)

```
/*
//LKED.A DD DSN=your library name (LIMDEP),DISP=SHR,SPACE=
//LKED.SYSIN DD *
  INCLUDE A(LIMDEP)
  ENTRY MAIN
/*
//GØ.SYSIN DD *
 (Your control and data pack(s)).
/*
```

Under this routine, you establish a new and temporary library called "A" which is not placed on disk, and which does not override your previously compiled LIMDEP program. An alternative, which is to use LKED.SYSLMOD DD card, keeps replacing LIMDEP on disk, with the newly compiled TRANS in place of the previous one. This adds CPU time during LKED, as well as using up your disk space, and necessitates your recompiling a dummy TRANS when you no longer want to use the previous one. The suggested method will be superior in most cases.

If you are using LIMDEP off of disk file, then your job package looks like Fig. 4:

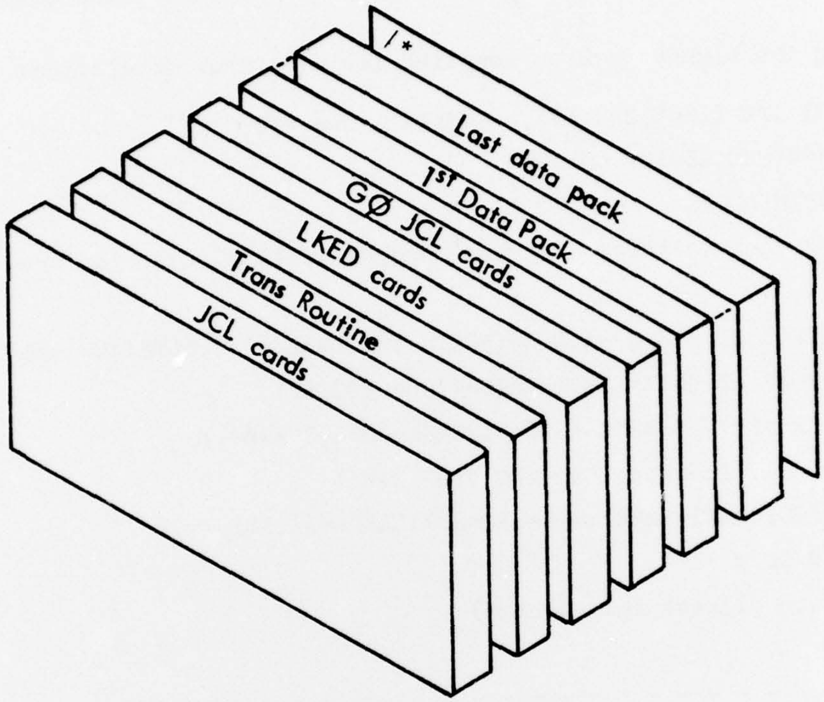


Fig. 4 —LIMDEP run - using disk library with new TRANS subroutine

1. Initial JCL cards, including JOB, and EXEC cards. If you are compiling a new TRANS, use EXEC FØRTCLG. If you are not adding a TRANS routine, just use EXEC FØRTLG. If you use FØRTCLG, be sure to include a //FØRT.SYSIN DD * card.
2. Your TRANS routine (OPTIONAL).
3. LKED cards--see section on "Running the Program on 370."
4. GØ step JCL cards.
5. Data packs. (See Fig. 2.)
6...Additional data packs (OPTIONAL).

SAMPLE JOB PACKAGES

(The following examples follow Rand convention--other users beware.)

1. Using the LIMDEP deck as supplied (or with your alterations to TRANS)

```
//P4560#01 JØB (1908,200,57), 'PHELPS', CLASS=B
// EXEC FØRTCLG,REGIØN.GØ=150K
//FØRT.SYSIN DD *
   (LIMDEP deck follows, with any changes in TRANS that you have made)
/*
//GØ.FT06F001 DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//GØ.FT09F001 DD UNIT=TEMP,SPACE=(7004,(80,6)),
//   DCB=(RECFM=VBS,LRECL=500,BLKSIZE=7004,BUFNØ=1)
//GØ.FT10F001 (identical to the FT09 card)
//GØ.FT11F001 UNIT=TAPE9,VØL=SER=007,LABEL=(1,SL)
//GØ.SYSIN DD *
  (Your data pack(s) follow here)
/*
```

- -

If the LIMDEP program is being used off a disk-library, the following JCL cards would be appropriate: (Assume that the library in which LIMDEP is stored is called M.M7805.A3748.ECØNLIB(LIMDEP)) Card input of data assumed.

```
//P4560#02 JØB (1908,200,57), 'PHELPS', CLASS=B
// EXEC FØRTCLG,REGIØN.GØ=150K
//FØRT.SYSIN DD *
  (Your revised TRANS routine goes here.)
/*
//LKED.A DD DSN=M.M7805.A3748.ECØNLIB(LIMDEP),DISP=SHR,SPACE=
//LKED.SYSIN DD *
  INCLUDE A(LIMDEP)
  ENTRY MAIN
/*
```

```
//GØ.FT06F001 DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//GØ.FT09F001 DD UNIT=TEMP,SPACE=(7004,(80,6)),
//GØ.FT10F001 (identical to the FT09 card)
//GØ.FT11F001 UNIT=TAPE9,VØL=SER=007,LABEL=(1,SL)
//GØ.SYSIN DD *
  (Your data pack(s) go here.)
/*
```

- -

If you are using LIMDEP off of disk, and you are not using a new TRANS, simply use the following procedure:

```
//P4560#03 JØB (1908,200,57),'PHELPS', CLASS=B
//JØBLIB DD DSN=M.M7805.A3748.ECØNLIB,DISP=SHR
// EXEC PGM=LIMDEP,REGIØN=150K
//GØ.FT05F001 DD DDNAME=SYSIN
```

(etc. the remaining cards are identical to example #1 for the GØ step)